Using semantic resources to improve a syntactic dependency parser

Schneider, Gerold

Abstract: Probabilistic syntactic parsing has made rapid progress, but is reaching a performance ceiling. More semantic resources need to be included. We exploit a number of semantic resources to improve parsing accuracy of a dependency parser. We compare semantic lexica on this task, then we extend the back-off chain by punishing underspecified decisions. Further, a simple distributional semantics approach is tested. Selectional restrictions are employed to boost interpretations that are semantically plausible. We also show that self-training can improve parsing even without needing a re-ranker, as we can rely on a sufficiently good estimation of parsing accuracy. Parsing large amounts of data and using it in self-training allows us to learn world knowledge from the distribution of syntactic relation. We show that the performance of the parser considerably improves due to our extensions.

Originally published at:
Schneider, Gerold (2012). Using semantic resources to improve a syntactic dependency parser. In: LREC 2012 Conference Workshop "Semantic Relations II", Istanbul, Turkey, 22 May 2012, 67-76.

Using semantic resources to improve a syntactic dependency parser

Gerold Schneider

Institue of Computational Linguistics, University of Zurich
gschneid@cl.uzh.ch

Abstract

Probabilistic syntactic parsing has made rapid progress, but is reaching a performance ceiling. More semantic resources need to be included. We exploit a number of semantic resources to improve parsing accuracy of a dependency parser. We compare semantic lexica on this task, then we extend the back-off chain by punishing underspecified decisions. Further, a simple distributional semantics approach is tested. Selectional restrictions are employed to boost interpretations that are semantically plausible. We also show that self-training can improve parsing even without needing a re-ranker, as we can rely on a sufficiently good estimation of parsing accuracy. Parsing large amounts of data and using it in self-training allows us to learn world knowledge from the distribution of syntactic relation. We show that the performance of the parser considerably improves due to our extensions.

Keywords: Exploitation of semantic resources for NLP applications, Syntactic parsing, WordNet and WordNet-like resources, Self-training, Distributional semantics

1. Introduction

Syntactic parsing has made impressive progress over the past decade. Still, performance even of the best parsers lags behind human performance considerably. Bi-lexical statistics (Collins, 1999) has led to a quantum leap in parsing performance. The interaction of lexis and grammar, as postulated by (Sinclair, 1991) or (Hunston and Francis, 2000), is exploited by bi-lexical statistics for the disambiguation task. In terms of psycholinguistics, prefabricated partial trees are recognized directly and usually not decomposed into subparts. In terms of semantics, lexical semantics is modeled as the distribution of grammatical relations between lexemes at the syntactic level and can be used to discover similar words (Lin, 1998) or WordNet synsets (Curran, 2004). (Grefenstette et al., 2011) present a compositional distributional model of meaning in vector space models (e.g. (Schütze, 1998)), where the semantic vector space of a word is defined in terms of its distributional syntax.

The performance of statistical parsers is now reaching a ceiling. Additional types of semantic resources need to be considered and included. We present experiments using an existing dependency parser and investigate the role of semantics for parser improvement in this paper. Two semantic lexica are compared for the reduction of data sparseness. We extend the backoff chain by punishing underspecified decisions. Further, a simple distributional semantics extension is tested. We then use selectional restrictions to boost interpretations that are semantically plausible. We also show that self-training can improve parsing even without using a re-ranker. Parsing large amounts of data and using it in self-training allows us to learn world knowledge from the distribution of syntactic relation.

1.1. The Pro3Gres parser

The parser used in this study, Pro3Gres (Schneider, 2008), is a Dependency parser. Its representation is very close to and can be mapped to GREVAL (Carroll et al., 2003) and the Stanford scheme (Haverinen et al., 2008).

The parser uses a hand-written *competence* grammar and a statistical *performance* disambiguation learnt from the Penn Treebank (Marcus et al., 1993). The parser uses a Maximum Likelihood Estimation (MLE) probability model for the bi-lexical performance disambiguation, which we briefly introduce here in preparation for the adaptations that we make in the paper. The parser estimates the probability of the dependency relation R at distance (in chunks) $dist$, given the lexical head a of the governor and the lexical head b of the dependent.

$$p(R, dist|a, b) = P(R|a, b) \cdot P(dist|R, a, b) \quad (1)$$

$$\cong \frac{\#(R, a, b)}{\#((\sum R), a, b)} \cdot \frac{\#(R, dist)}{\#R} \quad (2)$$

The assumption is taken that the distance depends only on the relation type, and that a relation is only ambiguous in terms of the relations with which it is in competition. In order to alleviate sparse data, the parser uses a back-off architecture similar to (Collins and Brooks, 1995), but it extends from PP-attachment to most of its dependency relations, and includes simple semantic classes from WordNet (Miller et al., 1990), as e.g. in (Merlo and Esteve Ferrer, 2006).

The MLE probability model and the backoffs differ slightly for some relations. We now describe the PP-attachment model, which uses tri-lexical disambiguation. PP-attachment is modeled as ambiguous between noun attachment and verb attachment (the latter including adjective attachment). It uses the putative parsing context of (Collins and Brooks, 1995) as an approximation, where every verb is in competition with one noun, and every noun is in competition with one verb. The actual competitions during parse time are never in direct comparison, but indirectly via the comparison of the putative parsing context.

An MLE probability is the result of the positive counts divided by the candidate counts. For the PP-attachment model, positive counts are all cases from the training corpus that do attach, and candidate counts are the cases that do attach *plus* cases that could attach but that do not, according to the putative parsing context. For verb attachment (the relation label is $pobj$), then, candidate cases are all cases

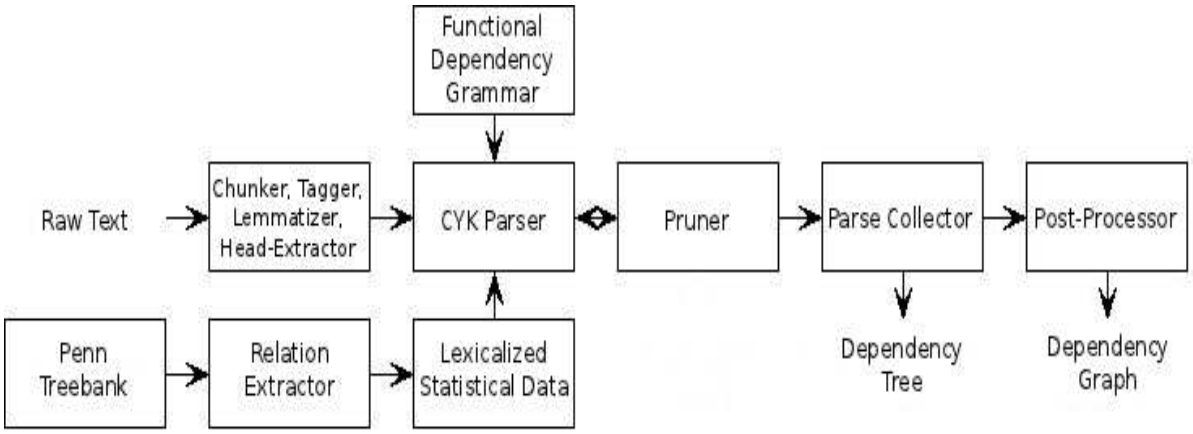

Figure 1: Pro3Gres flowchart

where attachment as $pobj$ occurs, *plus* all cases where in the ambiguous context of a verb-noun-PP sequence the PP attaches to the noun (the label is $modpp$).

$$p(pobj, dist|verb, prep, desc.noun) \cong \frac{\#(pobj,verb,prep,desc.noun)}{\#(pobj,verb,prep,desc.noun)+\#(modpp,verb,(\sum noun),prep,desc.noun)} \cdot \frac{\#(pobj,dist)}{\#pobj} \quad (3)$$

$$p(modpp, dist|noun, prep, desc.noun) \cong \frac{\#(modpp,noun,prep,desc.noun)}{\#(modpp,noun,prep,desc.noun)+\#(pobj,(\sum verb),noun,prep,desc.noun)} \cdot \frac{\#(modpp,dist)}{\#modpp} \quad (4)$$

(McDonald and Nivre, 2011) make a distinction between greedy, transition-based parsers like (Nivre, 2006) which take local decisions based on local state transitions (e.g. to shift or to reduce), and exhaustive graph-based parsers such as (McDonald et al., 2005) where (sub)graphs are modeled and many alternatives are kept. By their categorization Pro3Gres is an exhaustive graph-based parser. It uses a beam-search to discard unlikely partial analyses. Except for restrictions in the manually written grammar, the decisions of this parser are typically local. We will address this point in section 3.

The parser uses tagging and chunking as a preprocessing step, thus integrating fast finite-state techniques where appropriate, and converts dependency trees into graph structures in a post-processing step. The post-processing step includes the following incremental annotation: passive subjects are recognized, long-range dependencies are found, relative pronoun anaphora resolved, and verb-attached PPs are disambiguated between arguments and adjuncts.

An overview of the parser modules and their interactions is given in figure 1. We have chosen Pro3Gres for our experiments for the following reasons: (1) the strict separation into a manual grammar, which we have left unchanged, and a statistical disambiguation module is useful for our experiments, as it gives us control over the parameters, (2) as the parser uses explicit models and a restricted set of features it can be adapted fairly easily in order to conduct parsing experiments, (3) it shows a strong correlation between lexicalization and parsing quality, as we discuss in the following subsection.

1.2. The role of semantics for parsing

Bi-lexical statistics (Collins, 1999) has led to a quantum leap in parsing performance. But the debate on the importance of lexicalization is still open. On the one hand, decisions suffering from sparse data problems in the form of too little lexicalization lead to considerably worse results (e.g. (Collins and Brooks, 1995)), and approaches carefully extending lexicalisation can improve performance (McClosky et al., 2006; Stetina and Nagao, 1997). We have noticed a very strong correlation between backoff level and parser accuracy, as figure 2 illustrates for PP-attachment (nounpp = attachment of PP to a noun, verbpp=attachment of PP to a verb). Fully lexicalized decisions (Level 0: *head + preposition + description noun*), have much higher performance than those further down the back-off chain. Level 2 is *verb + preposition*, level 3 is *head class + preposition + noun*, level 4 is *verb class + preposition + description-noun class*, level 5 is *preposition + description-noun class*, level 6 is *preposition only*. We use the term description-noun to refer to the noun inside the PP.

On the other hand, (Gildea, 2001) have shown that monolexicalized approaches can perform almost as well. The approach of (Klein and Manning, 2003) is even unlexicalized; essentially it is an approach that uses semantic classes, stating that semantic classes can get one almost as far as pure bi-lexical preferences. One could tentatively summarize these opposing trends as follows: bi- and tri-lexicalized approaches can only perform well if data is not sparse, but data is sparse in the vast majority of cases. In those cases, a considerably less sparse good semantic classification can be as profitable. For this paper, it is tested in the following if there are semantics-based methods to reduce sparseness, so that more decisions can be taken at early backoff levels.

There are additional reasons why investigating the role of semantics for parsing is crucial. First, statistical approaches are now reaching a ceiling, although the error rate of even the best systems is still significantly and considerably higher than human inter-annotator disagreement. New sources of information need to be integrated. An obvious candidate for testing is semantics. Second, there are increasingly many approaches using syntactic modules for detection of thematic roles or doing syntactic parsing and thematic role detection simultaneously, see e.g. the CoNLL

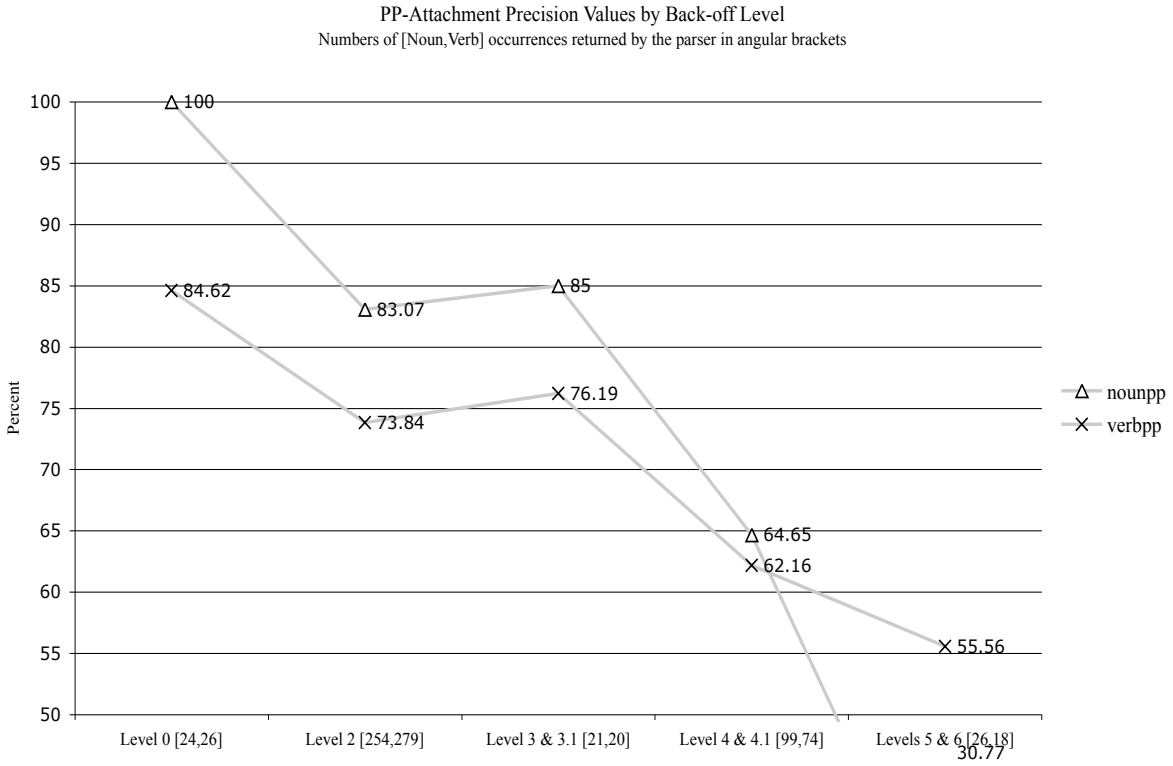

Figure 2: Evaluation: Quality of Backoff

2008 shared tass (Clark and Toutanova, 2008). Third, the types of errors that the various parsers make are often poorly understood. Investigating contributing factors, such as in (McClosky and Charniak, 2008) or even detailed error comparisons such as in (McDonald and Nivre, 2011) are very useful as they can help to disentangle lexical, syntactic and semantic factors.

In the rest of this paper, we will explore semantic factors to the end of increasing parsing performance. In section 2., we employ semantic information in the backoff system. In section 3., we use selectional restrictions and a non-local MLE model to boost plausible readings. We use semantic world-knowledge obtained from self-training in section 4. In section 5., we add an extension based on distributional similarity to the self-training model. Finally, we give an overview of the combined performance that we have gained due to our extensions in section 6.. We use GREVAL (Carroll et al., 2003) as evaluation corpus. It consists of 500 manually annotated sentences from the Susanne corpus.

2. Lexical semantic backoffs

We first report on experiments using semantic resources in the backoff.

2.1. Wordnet versus Levin class

We have discussed in the introduction that (Klein and Manning, 2003) have shown that a good semantic classification can get one as far as bi- and tri-lexicalized approaches. There are a number of semantic classification options for sparse data. We have used WordNet lexicographer file classes (Miller et al., 1990) as a simple approach, and alternatively Levin classes (Levin, 1993) for verbs. We compare the performance of these two ressources in figure 3. WordNet performs better in most cases. Also noun-PP attachment performance is indirectly affected. In order to break down performance across the whole confidence spectrum, we give threshold levels on the horizontal axis. The rightmost number, 0.9 means, for example, that only attachment decisions that were reported as being more than 90% probable in MLE attachment estimation (see introduction) were considered (which leads to high precision, but low recall). A potential reason why Levin classes perform worse is because their coverage is lower.

2.2. Similarity-based lexemes

We tested a number of extensions to fight the sparse data problem. In this section we employ an example-based use of the semantic constraints placed by syntactic relations. Because a head places strong selectional restrictions on its

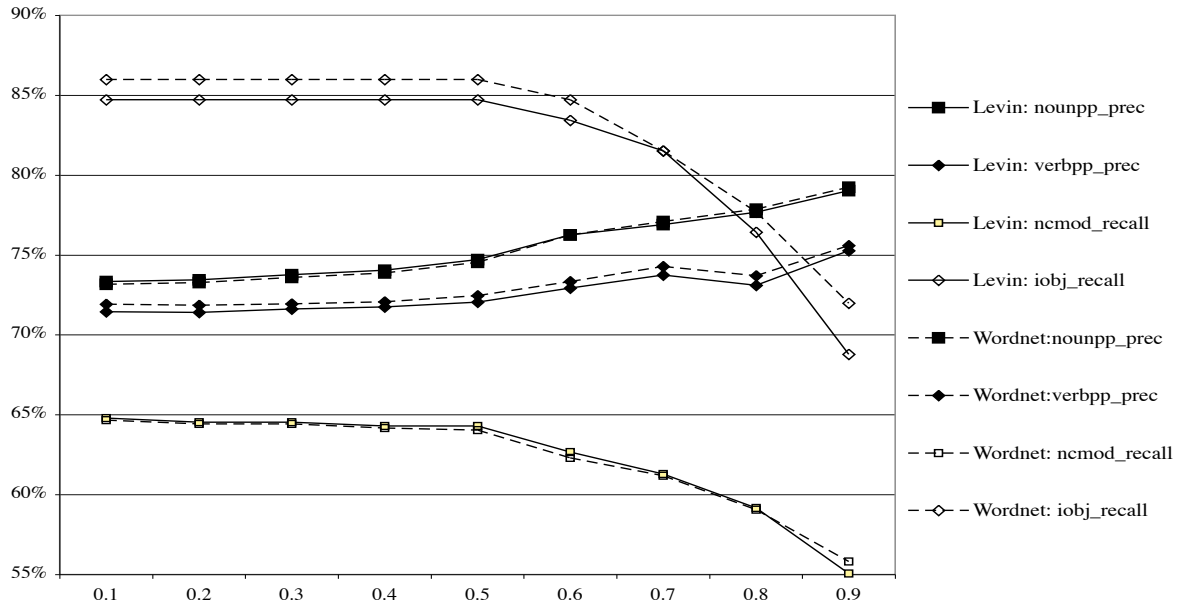

Figure 3: Comparison of Levin or Wordnet verb classes for backing off

dependent, dependents of the same head, or heads with the same dependent, are often similar. This fact can be exploited for Word Sense Disambiguation (e.g. (Lin, 1997)), the detection of similar words (Lin, 1998), WordNet synonyms (Curran, 2004) or distributional semantics vector models (Grefenstette et al., 2011). We use a very simple approximation here as follows:

For every target zero-count head-dependent pair, i.e. an attachment candidate at parse time for which we cannot find any occurrence at the first backoff level (the fully lexicalized level 0), if non-zero counts are found for both

1. a head'-dependent,
2. a head- dependent' and
3. a head'-dependent'

(where *head'* and *dependent'* are any word of the same tag as *head* and *dependent*, respectively), then their MLE counts are used. In a more restrictive version, only *dependent'* of the same WordNet noun class or verb class is allowed. Versions that use data from a large automatically parsed corpus (BNC) have also been tested. All of them show similar, slightly lower performance. An analysis of the decision points shows that non-zero values at between 2 and 10 times the original fully lexicalized level can be obtained, but the unreliability of the similarity and the increased coverage seem to level each other out. We assume that our first test was probably too simplistic. We will come back to this point again in section 5.

2.3. Unspecificity and probability

The level of backoff at which a decision can be taken is crucial as we have seen in figure 2. Better informed decisions are consistently better. At the first sight, informedness and probability seem unrelated. Informedness seems to have an impact on reliability and not on probability. On second thought, there is a reason why events are unseen – they are either indeed rare or simply impossible. The original parser only uses positive information. It also introduces artificial positive information in the form of smoothing, giving unseen events a low probability as is standardly done, but now we introduce positive information learning from the absence of word-word interactions.

From a probabilistic viewpoint, the negative information, although strictly speaking unquantifiable, that, whenever we can only decide late in the backoff chain, the fact that specific information is absent is an indirect indication that an event is indeed rare. In probability spaces where sparseness is relatively low, absence can be elevated to the status of partial evidence. If we had a complete system (closed world assumption), negative information (absence) could reliably be considered as positive information.

From a complementary distribution viewpoint, we have seen (figure 2) that there is a very strong relation between informedness expressed by the backoff level and performance. If a highly informed relation probability (say for verb PP-attachment) is in complementary distribution and hence competition with a less informed but equal probability (say for noun PP-attachment), we have evaluation performance statistics reasons to give preference to the highly informed relation.

PP-attachment	without "ironing"		with "ironing" (2%) = Base system	
subj_prec	849 of 946	89.75%	849 of 946	89.75%
local_subj_prec	826 of 912	90.57%	826 of 912	90.57%
subj_recall	855 of 1095	78.08%	855 of 1095	78.08%
obj_prec	353 of 412	85.68%	354 of 413	85.71%
obj_recall	351 of 428	82.01%	352 of 428	82.24%
nounpp_prec	351 of 497	70.62%	352 of 491	71.69%
verbpp_prec	353 of 477	74.00%	357 of 482	74.07%
ncmod_recall	530 of 801	66.17%	534 of 801	66.67%
iobj_recall	139 of 157	88.54%	140 of 157	89.17%
argmod_recall	34 of 40	85.0%	35 of 40	87.5%

Table 1: Results of evaluation with and without "ironing". Ironing takes unspecificity as expressed by backoff level as a punishing factor, we have used two 2% lower probability per backoff level

From a post-hoc performance perspective, there should be some way of taking the actual performance that is to be expected into consideration. With the benefit of hindsight, seeing that such an approach performs better, it makes sense to counter-balance obvious tendencies.
Although its status is probabilistically unclear, we have experimented with a simple extension for the PP-attachment relations that introduces an unspecificity punishment factor into the probability calculation. In our example, each probability is reduced by 2 percent for each backoff step. The results for some of the most frequent relations are given in table 1. Except for the subject relation, every relation shows an increase both in precision and in recall. The ambiguous PP-attachment relations profit in particular. The exact meaning of the labels is as follows:

- subj_prec , subj_recall: Precision and recall of the subject relation
- local_subj_prec: Precision of subject that are not in a long-distance relation, i.e. that are overtly expressed
- obj_prec , obj_recall: Precision and recall of the object relation
- nounpp_prec: Precision of the noun-PP attachment relation $modpp$
- verbpp_prec: Precision of the verb-PP attachment relation $pobj$
- ncmod_recall: Recall of PP adjuncts (mostly nominal, i.e. $modpp$)
- iobj_recall: Recall of PP arguments (mostly verbal, i.e. $pobj$)
- argmod_recall: Recall of *by*-agents in passive clauses (a part of $pobj$)

In distinction to smoothing, where positive information is produced, one could call this method *ironing*, because negative information irons out unwarranted and unjustified creases of too high probability caused by underspecificity. With values between 1 and 5%, "ironing" leads to better results, with values above that, results decline again. We use the model with 2% ironing as our base system for the following sections.
It has been shown for the fields of unsupervised grammar induction (Smith and Eisner, 2005) and for document classification (Schneider, 2004) that the ability of the classifier to use negative evidence makes a crucial difference in terms of performance.

3. Semantic Restrictions

In this section, we use selectional restrictions and a non-local MLE model to boost plausible readings.

3.1. Selectional Restrictions

We have discussed in section 1 that the original parser models probabilities using only those syntactic relations that are in competition. For example, every verb is in competition with one noun, the fact that several nouns may be in competition in a stacked NP is not modeled directly. Similarly, objects (e.g. *eat pizza*) and nominal adjuncts (e.g. *eat Friday*) are modeled as being in competition, but not subjects and objects. One could say that the original parser strictly models syntactic competition, to which we now add semantic competition. In the additionally introduced semantic probability model, every relation is in competition with every other relation. In order to calculate the probability for a verb-object relation between *rabbit* and *chase* we use the general probability of verb-object relation between *rabbit* and *chase* irrespective of which relations the object relation is in competition with. This has the effect that, in all likelihood, a sentence like *the rabbit chased the dog* gets a lower probability then *the dog chased the rabbit* because rabbits are very unlikely to be subjects of active instances of *chase*. Thus, our semantic world knowledge becomes part of the model, the parser parses for what is semantically more plausible. We will refer to this model as selectional restriction. While such an approach entails the risk of misinterpreting surprising new information, it is also psycholinguistically adequate: human parsers often disambiguate by using their expectations and their world knowledge. The results of the selectional restrictions model are given in table 2. The performance of almost every relation increases or stays unchanged.

3.2. Non-local Decisions

We have discussed in the introduction that the probabilities of the Pro3Gres parser are local, which means that world-knowledge expressed across more than one node generation is lost in the model. Although locality extends further in Dependency Grammar than in constituency grammar (where trees are more nested) and although there are global restrictions in the hand-written grammar, this is a serious shortcoming. In stacked PPs, for example, in the sequence verb-PP_1-PP_2 the attachment probabilities for verb-PP_1, verb-PP_2, and PP_1-PP_2 are only considered independently. It is well known that considering sister, grandmother and great-grandmother nodes increases parsing accuracy (e.g. (Charniak, 2000), (Bod et al., 2003)), particularly in the case of the highly ambiguous PP-attachment

Relation	without sel. rec. = Base system		with sel. rec.	
subj_prec	849 of 946	89.75%	854 of 950	89.89%
local_subj_prec	826 of 912	90.57%	830 of 916	90.61%
subj_recall	855 of 1095	78.08%	860 of 1095	78.54%
obj_prec	354 of 413	85.71%	354 of 414	85.51%
obj_recall	352 of 428	82.24%	352 of 428	82.24%
nounpp_prec	352 of 491	71.69%	353 of 486	72.63%
verbpp_prec	357 of 482	74.07%	358 of 480	74.58%
ncmod_recall	534 of 801	66.67%	535 of 801	66.79%
iobj_recall	140 of 157	89.17%	140 of 157	89.17%
argmod_recall	35 of 40	87.5%	35 of 40	87.5%

Table 2: Results of evaluation with and without selectional restrictions

PP-attachment	without multi-PP = Base system		with multi-PP	
nounpp_prec	352 of 491	71.69%	354 of 492	71.95%
verbpp_prec	357 of 482	74.07%	357 of 481	74.22%
ncmod_recall	534 of 801	66.67%	536 of 801	66.92%
iobj_recall	140 of 157	89.17%	140 of 157	89.17%
argmod_recall	35 of 40	87.5%	35 of 40	87.5%

Table 3: Results of evaluation with and without stacked PP model

relations. We have therefore added an MLE model which calculates the probabilities for verb-PP_1-PP_2 sequences and noun-PP_1-PP_2 sequences. For example, the probability that PP_2 is a dependent of PP_1 ($PP_1 < PP_2$) in a verb-PP-PP sequence, given the lexical items, is calculated as follows:

$$p(verb < (PP_1 < PP_2)) = \frac{\#(verb<(PP_1<PP_2))}{\#(verb<(PP_1<PP_2))+\#((verb<PP_1)<PP_2))}$$

The data is so sparse that in most cases only backoffs where all verbs and noun are replaced by their semantic verb- and noun-classes from Wordnet deliver results. The performance of the base system is compared to the new model in table 3, showing a slight improvement.

4. Distributional Semantics: Self-Training

The use of large amounts of parsed data is known as *self-training*. The variance of a large corpus is so big that it gives an opportunity to learn from the several different configurations, and parsing results from the many configurations with relatively low ambiguity may deliver a signal that is strong enough. In a nutshell, self-training can improve results where sparseness is worse than error rate. From a semantic viewpoint, parsing large amounts of data allows us to learn world knowledge from the distribution of syntactic relations. The main danger of self-learning is that the ensuing corpus skew will lead to the same problems as in co-training (Sarkar, 2001) and boost errors. Until recently, self-training was thought to be unable to lead to better performance (Charniak, 1997; Steedman et al., 2003). (Bacchiani et al., 2006) have shown that self-training can improve parsing out-of-domain texts, and is therefore a suitable approach for domain adaptation. (McClosky et al., 2006) was the first approach to show that the use of a re-ranker (Charniak and Johnson, 2005) can also improve in-domain parsing. Their re-ranker uses a very rich set of features, which leads to a sufficiently different view on the data to allow for an increase in performance.

(McClosky et al., 2008) describe some of the reasons that lead to an improvement from self-training. They reject the assumptions that high performance of the underlying parser is a prerequisite and that analyses that are missed by the underlying parser are a problem. They find out that two major sources of improved performance are (1) the different view on the data and (2) the reduction of sparseness: bi-lexical heads that are unseen in the Penn Treebank but seen in the self-training lead to a clear improvement: *"H (biheads) is the strongest single feature and the only one to be significantly better than the baseline"* (p. 567). This indicates that the debate on the importance lexicalization is still open.

A reliable measure of confidence on whether a parser decision is correct or not plays a crucial role in self-training. If this measure were completely reliable, only correct parses would be added to the training corpus. The parser which we use offers a sufficiently good measure: there is a very strong correlation between backoff level and the correctness of the parser decision, as figure 2 shows. This can be exploited, e.g. by adding self-training results late in the backoff chain, thus using tri- or bi-lexical self-training decisions if the Penn Treebank training data only offers monolexical decisions.

The Penn Treebank contains 1 million words. We have parsed the 100 million words British National Corpus BNC (Aston and Burnard, 1998), which gives us 2 orders of magnitude more lexicalized data to alleviate the sparse data problem. The PP-attachment error rate on the BNC is clearly lower than the error rate on PP-attachment cases from low backoff-levels (figure 2). We have added the self-trained counts into the backoff hierarchy between level 2 and 3. The results are given in table 4. There is a small increase in the PP-attachment relations. The increase is too small to be statistically significant, however, so it can only serve as an indication. Therefore, a larger evaluation corpus will be needed. There are only 43 cases in GREVAL in which the top-ranked reading includes a decision from the new self-trained backoff level, which means that we obtain 3 improvements out of 43 cases.

Most approaches to self-training use a re-ranker, e.g. (McClosky et al., 2006) as a crucial element. We have presented an approach which does not need a re-ranker but improves performance. It is known that co-training (Sarkar, 2001; Hwa et al., 2003) only leads to minimal improvements. Our approach is different from co-training for a number of reasons: (1) for highly informed levels, we only use the original training set, and (2) we retain all parses, which reduces the risk of skewing the corpus or disappearing into an "error hole" as it can typically happen in co-training.

Relation	without BNC self = Base system		with BNC self	
subj_prec	849 of 946	89.75%	849 of 946	89.75%
local_subj_prec	826 of 912	90.57%	826 of 912	90.57%
subj_recall	855 of 1095	78.08%	855 of 1095	78.08%
obj_prec	354 of 413	85.71%	354 of 413	85.71%
obj_recall	352 of 428	82.24%	352 of 428	82.24%
nounpp_prec	352 of 491	71.69%	353 of 492	71.75%
verbpp_prec	357 of 482	74.07%	357 of 481	74.22%
ncmod_recall	534 of 801	66.67%	534 of 801	66.67%
iobj_recall	140 of 157	89.17%	140 of 157	89.17%
argmod_recall	35 of 40	87.5%	36 of 40	90.0%

Table 4: Results of evaluation with and without self-training

5. Combining self-training and example-based similarity

We have learnt in the previous section that self-learning can work if we have a reasonably reliable measure indicating where sparse data leads to errors. Such a measure can be obtained from the backoff level, and thus we use self-training decisions only for late backoff instances. We have learnt in section 2 that simplistic "naive" approaches to distributional similarity do not work. We have used similarity-based counts directly after the the fully lexicalized level 0. The imprecision that such a simplistic similarity approach introduces is probably still higher than the error rate at the second-highest backoff level. We thus re-delegate the similarity-based approach to the level after the BNC-self-trained data. The data from the parsed BNC is used, and the restrictive version, in which only *head'* and *dependent'* of the same WordNet noun class or verb class as *head* and *dependent*, respectively, is allowed. Performance is very similar to the self-trained model in the previous section.

We have made a further restrictions: similarity-pairs (*head'-dependent, head'-dependent'* and *head'-dependent'*) are generated from the BNC, but only MLE probabilities from the error-free Penn Treebank are allowed, i.e. if the Penn treebank contains data for a *head'-dependent* or *head-dependent'* pair it is taken, otherwise the backoff chain continues resorting to the next, lower level. Results are given in table 5, comparing the self-trained model to the self-trained similarity model. We have added this extension only to the PP-attachment relations. Again, the improvement is probably strictly speaking not statistically significant. In the GREVAL corpus, there are 7 cases that improve. There are only 13 cases, however, in which the top-ranked reading includes a decision from the new self-trained plus similarity backoff level, which means an improvement of 7 out of 13.

We would like to use a vector-based semantics model in future research, for example (Grefenstette et al., 2011). The current pilot study has shown that a gain in parsing performance from using similarity-based metrics against sparse data can be expected.

Relation	BNC self =right col. of table 4		BNC self + similarity	
nounpp_prec	353 of 492	71.75%	356 of 494	72.06%
verbpp_prec	357 of 481	74.22%	357 of 479	74.53%
ncmod_recall	534 of 801	66.67%	538 of 801	67.17%
iobj_recall	140 of 157	89.17%	140 of 157	89.17%
argmod_recall	36 of 40	90.0%	36 of 40	90.0%

Table 5: Results of evaluation with original self-training and with added example-based similarity

Relation	Base System		Combined	
subj_prec	849 of 946	89.75%	854 of 950	89.89%
local_subj_prec	826 of 912	90.57%	830 of 916	90.61%
subj_recall	855 of 1095	78.08%	860 of 1095	78.54%
obj_prec	354 of 413	85.71%	354 of 414	85.50%
obj_recall	352 of 428	82.24%	352 of 428	82.24%
nounpp_prec	352 of 491	71.69%	359 of 491	73.12%
verbpp_prec	357 of 482	74.07%	357 of 475	75.16%
ncmod_recall	534 of 801	66.67%	541 of 801	67.54%
iobj_recall	140 of 157	89.17%	140 of 157	89.17%
argmod_recall	35 of 40	87.5%	35 of 40	87.5%

Table 6: Evaluation comparison between base system and combined additions

6. Combined Model and Discussion

Finally, we give an overview of the combined performance that we have gained from the extensions introduced in sections 3 to 5. The results are given in table 6. Performance remains unchanged in 3 lines, there is one slight decline (*obj* recall), PP-attachment precision increases by over a percent, while recall also slightly improves. In terms of parsing speed, the extensions made in sections 4 and 5 are costly. The original parser parses the 500 sentence GREVAL corpus in under a minute, and the 100 million words BNC in about a day. Parsing times in sections 2 and 3 hardly change, in section 4 it increases to about a minute and to about 5 minutes in section 5.

While a performance increase of maximally 1.5% may seem very moderate, it should be considered in view of the law of diminishing marginal utility, in comparison to the baseline and the upper bound, and supplemented with an analysis of errors. For this discussion, we will focus on the PP-attachment relations.

As a PP-attachment baseline model, we use a version of the parser that uses the base system for all relations, but for the PP-attachment relations it only uses the preposition, i.e. backoff level 6. Results are given in table 7, first column (*Baseline*). In terms of precision, the increase from the base system to the combined system is as big as the one from baseline to base system, about 1.4%. In terms of recall, the increase from the baseline to the base system is 2.4%, the increase from the base system to the combined system is another 0.7%.

As PP upper bound, we use version of the combined system that reports not only the top ranked, but the first 64 readings for every sentence. While precision is negatively affected by a random element, the recall thus obtained gives one an

Relation	Baseline		Base System		Combined		Upper Bound	
nounpp_prec	337 of 472	71.40%	352 of 491	71.69%	359 of 491	73.12%	–	–
verbpp_prec	358 of 501	71.46%	357 of 482	74.07%	357 of 475	75.16%	–	–
ncmod_recall	517 of 801	64.54%	534 of 801	66.67%	541 of 801	67.54%	630 of 801	78.65%
iobj_recall	139 of 157	88.54%	140 of 157	89.17%	140 of 157	89.17%	144 of 157	91.71%
argmod_recall	39 of 40	97.50%	35 of 40	87.50%	35 of 40	87.50%	40 of 40	100%
$\sum$ PP Prec	695 of 973	71.43%	709 of 973	72.87%	716 of 966	74.12%	–	–
$\sum$ PP Recall	695 of 998	69.64%	709 of 998	71.04%	716 of 998	71.74%	814 of 998	81.56%

Table 7: Evaluation comparison for PP-attachment relations between baseline, base system, combined additions and upper bound

Relation	Attachment Error	Head Extraction Error	Chunking or Tagging	compl/prep Error	Grammar Mistake or incompl. Parse	Grammar Assumption
Noun-PP Precision	22	1	8	0	3	3
Noun-PP Recall	25	1	14	0	12	5
Verb-PP Precision	12	1	5	1	1	2
Verb-PP Recall	2	0	1	0	0	0
Totals	61	3	28	1	16	10
Proportions	51 %	3 %	24 %	1 %	13 %	8 %

Table 8: Detailed Analysis of the PP-attachment errors in the first 100 evaluation corpus sentences

assessment of the how accurate results can get if an oracle ranked all possible readings correctly. The recall measures are given in table 7, last column (*Upper Bound*), showing that the 1% improvement in $ncmod\_recall$ corresponds to almost a tenth of the maximally possible increase.

An analysis of PP-attachment errors in table 8 shows why almost a fifth of $ncmod$ cannot be found. We have investigated the PP-attachment errors in the first 100 sentences in the 500 sentence evaluation corpus (GREVAL, (Carroll et al., 2003)) in (Schneider, 2008), according to the output of the base system. About half of the errors are attachment errors, almost a quarter are chunking or tagging errors. Grammar mistakes or incomplete parses are cases which the grammar did not handle correctly, for example because the grammar does not allow X-bar violations and places strong restrictions PPs that precede their governor. The category of grammar assumption involves cases where our intended analysis as mirrored in our grammar does not coincide with the grammar view of the gold standard annotators. The majority of attachment errors can be corrected by selecting the correct non-first analysis, other errors cannot be corrected by our current parser.

7. Conclusion

We have successfully used several semantic resources to improve the performance of a syntactic dependency parser and have learnt a number of things on the way. We have learnt in section 2 that our first very simple approach to using similarity-based measures does not improve performance. We have learnt that Levin classes lead to a smaller improvement than WordNet classes. We have seen that negative information can up to a point be used as partial evidence. Although its probabilistic status is unclear, punishing late backoff decisions considerably improves performance. We have called our approach *ironing* because negative information irons out unwarranted and unjustified creases of too high probability caused by underspecificity.

In section 3, we have employed selectional restrictions to boost interpretations that are semantically plausible. We have also added an MLE model considering grandmother and sister node information for PP attachment in order to be able to profit form world knowledge that is expressed across two node generations. Both extensions increase performance.

In section 4, we have presented an approach using self-training which does not need a re-ranker, unlike e.g. (McClosky et al., 2006), and shown that it leads to improved performance. We use a parser which delivers a relatively reliable measure of parsing quality (figure 2), which we can exploit. We have learnt that self-training can work if we apply it only in those cases where we know that the expected backoff performance is lower than general parser performance.

In section 5, we use what we have learnt in section 4 to improve our simple distributional semantics approach to detect similar words. If we constrain our criteria to detect similar words, use only MLE counts from the Penn Treebank, and add the model late in the backoff chain (where decisions are of relatively poor quality) we gain a considerable improvement in parsing quality.

Finally, we combine the improvements made in sections 3 to 5. Particularly the ambiguous PP-attachment relations improve. PP-attachment precision improves by over 1% while also recall improves slightly. We discuss the performance in comparison to a baseline and the upper bound and give a brief error analysis.

An additional conclusion that we can draw from the current pilot study is that employing semantic resources has the potential to increase the performance of parsers considerably. More systematic approaches, for example using

vector-space models (Grefenstette et al., 2011) and large evaluation corpora will be used in future research.

8. References

Guy Aston and Lou Burnard. 1998. *The BNC Handbook. Exploring the British National Corpus with SARA*. Edinburgh University Press, Edinburgh.

Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Rens Bod, Remko Scha, and Khalil Sima'an, editors. 2003. *Data-Oriented Parsing*. Center for the Study of Language and Information, Studies in Computational Linguistics (CSLI-SCL). Chicago University Press.

John Carroll, Guido Minnen, and Edward Briscoe. 2003. Parser evaluation: using a grammatical relation annotation scheme. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 299–316. Kluwer, Dordrecht.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proc. of the 15th Annual Conference on Artificial Intelligence (AAAI-97)*, Stanford, USA.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North American Chapter of the ACL*, pages 132–139.

Alexander Clark and Kristina Toutanova, editors. 2008. *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Coling 2008 Organizing Committee, Manchester, England, August.

Michael Collins and James Brooks. 1995. Prepositional attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Doctoral thesis, Institute for Communicating and Collaborative Systems, University of Edinburgh.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202, Pittsburgh, PA.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, Oxford.

Katri Haverinen, Filip Ginter, Sampo Pyysalo, and Tapio Salakoski. 2008. Accurate conversion of dependency parses: targeting the Stanford scheme. In Tapio Salakoski, Dietrich Rebholz-Schuhmann, and Sampo Pyysalo, editors, *Proceedings of Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 133–136, Turku, Finland. Turku Centre for Computer Science (TUCS).

Susan Hunston and Gill Francis. 2000. *Pattern Grammar: A Corpus-Driven Approach to the Lexical Grammar of English*. Benjamins, Amsterdam/Philadelphia.

Rebecca Hwa, Miles Osborne, Anoop Sarkar, and Mark Steedman. 2003. Corrected co-training for statistical parsers. In *Proceedings of the ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining at the 20th International Conference on Machine Learning (ICML-2003)*, Washington DC.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

Beth C. Levin. 1993. *English Verb Classes and Alternations: a Preliminary Investigation*. University of Chicago Press, Chicago, IL.

Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid. Association for Computational Linguistics.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL '98)*, pages 768–774, Montreal.

Mitch Marcus, Beatrice Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of ACL-08: HLT, Short Papers*, pages 101–104, Columbus, Ohio, June. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344, Sydney, Australia, July. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2008. When is self-training effective for parsing? In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 561–568, Manchester, UK, August. Coling 2008 Organizing Committee.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguis-*

tics, 37(1):197–228.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Paola Merlo and Eva Esteve Ferrer. 2006. The notion of argument in PP attachment. *Computational Linguistics*, 32(2):341 – 378.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.

Joakim Nivre. 2006. *Inductive Dependency Parsing*. Text, Speech and Language Technology 34. Springer, Dordrecht, The Netherlands.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of NAACL 2001*, Pittsburgh, PA.

Karl-Michael Schneider. 2004. On word frequency information and negative evidence in naive bayes text classification. In *Proceedings of España for natural language processing, ESTAL*.

Gerold Schneider. 2008. *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

John Sinclair. 1991. *Corpus, Concordance, Collocation*. OUP, Oxford.

Noah A. Smith and Jason Eisner. 2005. Guiding unsupervised grammar induction using contrastive estimation. In *International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Grammatical Inference Applications*, pages 73–82, Edinburgh, July.

Mark Steedman, Steven Baker, Jeremiah Crim, Stephen Clark, Julia Hockenmaier, Rebecca Hwa, Miles Osborne, Paul Ruhlen, and Anoop Sarkar. 2003. Semi-supervised training for statistical parsing. Technical Report CLSP WS-02 Final report, John Hopkins University.

Jiri Stetina and Makoto Nagao. 1997. Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the Fifth Workshop on Very Large Corpora*, pages 66–80, Beijing and Hong Kong, Aug. 18 – 20.